REMARKS

Claims 1-3, 5-11, 15, 22-23 were examined. No claims are cancelled. Claims 1, 2, and 7-11 are requested to be amended. Claims 33-42 are requested to be added. Accordingly, claims 1-3, 5-11, 15 and 22-42 remain in the application.

Attached hereto is a marked-up version of the changes made to the claims by the current amendment. The attachment is captioned "Version With Markings To Show Changes Made."

I. Claims Rejected Under 35 U.S.C. § 103

Claims 1-3, 5-8, 11, 22-28 and 31 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,167,538 issued to Neufeld, et al. (hereinafter "Neufeld") in view of U.S. Patent No. 6,311,056 issued to Sandidge (hereinafter "Sandidge"). Applicants respectfully disagree for the following reasons.

In regard to claims 1 and 22, Neufeld does not teach a real time operating system (RTOS), performance monitoring driver, or a real time operating system of an Input/Output (I/O) processor. The Examiner relies upon Neufeld to teach an RTOS and equates the operating system 108 in Figure 1 taught by Neufeld to a real time operating system. However, the Examiner has not indicated any part of Neufeld that teaches or suggests that the operating system 108 in Figure 1 and discussed at col. 4, lines 49-57 is a real time operating system. One of ordinary skill in the art would understand that the term operating system used in the context in Neufeld does not encompass a real time operating system. Real time operating systems have a set of characteristics and requirements that set them apart from the conventional operating system discussed in Neufeld. Real time operating systems have unique requirements related to determinism, responsiveness, user control, reliability and fail-soft operation. See Exhibit A, Operating Systems: Internal and Design Principles by William Stallings, Prentice-Hall, Inc. 1998, pages 430-432. The network operating system by Novell, Inc. which is cited in Neufeld as an exemplary operating system for operating system 108 is an example of a conventional operating system. See col. 4, lines 55-57.

Thus, Neufeld does not teach or suggest a method involving a real time operating system as claimed in claims 1 and 22.

Further, Neufeld does not teach or suggest a real time operating system operating on an I/O processor. Rather, Neufeld teaches that the operating system 108 is run by the host processor 616, not the keyboard controller 626 which the Examiner has identified as the equivalent of an I/O processor. One of ordinary skill in the art would understand that the keyboard controller of Neufeld is of insufficient complexity to support an operating system or RTOS. Thus, Neufeld does not teach an RTOS running on an I/O processor.

Further, Neufeld does not teach sending selected events for which information is to be gathered on the I/O processor as message request from the host processor to the RTOS on the I/O processor. Rather, Neufeld teaches that the microprocessor 616 performs the monitoring of the various components of the computer system. See col. 10, lines 67 to col. 11, line 3 of Neufeld. The Examiner has failed to identify any part of Neufeld that teaches or suggests an I/O processor running an RTOS that receives message requests from a host processor as to what data is to be gathered. Instead, the Examiner has equated a keyboard controller with an I/O processor but has failed to indicate any part of Neufeld that teaches that this keyboard controller runs an RTOS that receives messages from the first processor as to what data is to be gathered. Performance monitoring is accomplished by the microprocessor 616 in Figure 6 which as taught by Neufeld sends messages to and from receivers messages drivers of resources being monitored. See Neufeld, col. 7, lines 21-35. This teaches away from the use of an RTOS with an I/O processor that registers a performance monitor driver and which receives messages of selected events for data to be gathered because the role of monitoring data and communicating with the components being monitored is performed by the host processor which runs the performance monitoring software.

Sandidge does not cure the defects of Neufeld. The Examiner has not indicated any part of Sandidge that teaches or suggests a real time operating system operating on an I/O processor that registers a performance monitoring driver which receives messages containing selected events to be monitored from a host processor. In relying on Sandidge for teaching the translation of a message

request into parameters, the Examiner combines unrelated prior art and fails to consider the claim as a whole. Claim 11 as a whole claims translating a message including selected events related to an I/O processor into parameters that are accessible to a real time operating system of an I/O processor. The Examiner has not indicated any part of Sandidge that teaches or suggests these elements of claim 1. Rather, the cited sections of Sandidge relate to the passing of encoded parameters between various elements of a cellular network. The cited sections of Sandidge do not teach translations of message requests into a set of parameters that are accessible to a real time operating system. The cited sections do not discuss a real time operating system. Therefore, Neufeld, in view of Sandidge does not teach or suggest each of the elements of claims 1 and 22. Accordingly, reconsideration and withdrawal of the obviousness rejection of claims 1 and 22 are requested.

In regard to claims 2-4, 6-15, 23-28 and 31, these claims depend from independent claims 1 and 22 respectively and incorporate the limitations thereof. Therefore, these claims are not obvious over Neufeld in view of Sandidge at least for the reasons mentioned in regard to claim 1.

In regard to claim 5, this claim depends from independent claim 1 and incorporates the limitations thereof. Thus, for the reasons mentioned in regard to claim 1, this claim is not obvious over Neufeld in view of Sandidge. Further, the Examiner has failed to indicate any part of Neufeld or Sandidge that teaches or suggests sending the message request through an operating system specific module of a host processor. Accordingly, reconsideration and withdrawal of the obviousness rejection of claim 5 are requested.

Claims 9, 10, 15, 29, 30 and 32 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Neufeld, in view of Sandidge and further in view of U.S. Patent No. 6,052,694 issued to Bromberg (hereinafter "Bromberg"). Applicants respectfully disagree for the following reasons.

In regard to claims 9, 10, 29 and 30, these claims depend from independent claims 1 and 22 and incorporate the limitations thereof. Thus, for the reasons mentioned in regard to claims 1 and 22, these claims are not obvious over Neufeld in view of Sandidge. Further, the Examiner

admits that Neufeld and Sandidge do not teach generating performance monitoring storage tables within the memory of an I/O processor. Bromberg is cited as teaching generating performance monitoring storage tables within the memory of an I/O processor. However, the Examiner has failed to indicate any part of Bromberg that teaches or suggest storing performance data within the memory of an I/O processor. Rather, as illustrated in Figure 2, Bromberg teaches a database that operates on a conventional computer system running conventional operating systems such as MS DOS, Windows™ or Unix. See col. 3, lines 39-50. Thus, the Examiner has failed to indicate any part of Bromberg that teaches a means for storing performance data in an I/O processor memory. Therefore, each of the elements of claims 9, 10, 29 and 30 are not taught or suggested by Neufeld in view of Sandidge and further in view of Bromberg. Accordingly, reconsideration and withdrawal of the obviousness rejection of these claims are requested.

In regard to claims 15 and 32, these claims depend from independent claims 1 and 22 and incorporate the limitations thereof. Therefore, these claims are not obvious over Neufeld in view of Sandidge. Further, the Examiner has failed to indicate any part of Bromberg that correct the defects of Neufeld and Sandidge. Therefore, claims 15 and 32 are not obvious over Neufeld in view of Sandidge and further in view of Bromberg. Accordingly, reconsideration and withdrawal of the obviousness rejection of claims 15 and 32 are requested.

II. New Claims

Claims 33-42 are requested to be added. These claims include many of the elements claimed in claims 1 and 12 and are patentable for the reasons mentioned above.

CONCLUSION

In view of the foregoing, it is believed that all claims now pending, namely claims 1-3, 5-11, 15 and 22-42 patentably define the subject invention over the prior art of record, and are in condition for allowance and such action is earnestly solicited at the earliest possible date. If the Examiner believes that a telephone conference would be useful in moving the application forward to allowance, the Examiner is encouraged to contact the undersigned at (310) 207 3800.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: 11/5, 2002

Eric S. Hyman, Reg. No. 30,139

12400 Wilshire Blvd.
Seventh Floor
Los Angeles, California 90025
(310) 207-3800

CERTIFICATE OF TRANSMISSION:

I hereby certify that this paper is being facsimile transmitted to the U.S. Patent and Trademark Office on November 5, 2002.

Lillian E. Rodriguez 11-5-02

November 5, 2002

FAX RECEIVED

NOV 0 5 2002

TECHNOLOGY CENTER 2800

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE CLAIMS

Please amend the claims as follows:

1. (Amended Four Times) A method comprising:

registering a performance monitoring driver [as a private driver] with a real time operating system (RTOS) of an input/output (I/O) processor, wherein the performance monitoring driver is coupled to a performance monitoring unit (PMU);

selecting events within the I/O processor to gather data on;

sending the selected events as a message request from a host processor to the RTOS of the I/O processor; and

translating the message request into parameters [based on a set of private group parameters] that are accessible by the RTOS.

2. (Amended Four Times) The method of claim 1, further comprising:

sending the message request as a translated request to the PMU[;

returning the pieces of data requested by the translated request to the performance monitoring driver; and

sending the pieces of data to a location specified in the message request].

7. (Amended) The method of claim 1, wherein the [set of private group] parameters includes at least one of (i) control parameters for hardware-based performance monitoring resources, (ii) mode-specific control parameters for a performance monitoring resource, and (iii) data parameters for at least one mode in one counter.

8. (Twice Amended) The method of claim 7, wherein the [set of private group] parameters includes at least one of the following parameters: an adjusted sample, a control, a counter, a current mode, a current time, an ending time, an interval, a lock control, a maximum algorithm, a maximum mode, a minimum sample interval, a minimum sample unit, a mode control, a number

counter, type of performance monitoring hardware available, a sample interval, a sigma time, and a status.

9. (Amended Four Times) The method of claim 1, further comprising:

generating performance monitoring storage tables within a memory of the I/O processor.

10. (Amended) The method of claim [9]33, further comprising:

subsequent to returning the [pieces of] data requested by the translated request to the performance monitoring driver, sending the [pieces of] data to the performance monitoring storage tables.

11. (Three Times Amended) The method of claim [2]33, wherein sending the [pieces of] data to a location specified in the message request further includes sending the [pieces of] data at a time period specified in the message request.